First Hit Fwd Refs

Generate Collection

L20: Entry 14 of 17 File: USPT May 10, 1994

DOCUMENT-IDENTIFIER: US 5311516 A
TITLE: Paging system using message fragmentation to redistribute traffic

Application Filing Date (1):
19921123

Detailed Description Text (4):
When a call to place a message, i.e., a paging request, is received, a controller 204 handles the processing of the message. The controller 204 is preferably a microcomputer, such as one based on the MC68000 family, which is manufactured by Motorola Inc., or the equivalent. The controller 204 runs various pre-programmed routines for controlling such terminal operations as voice prompts to direct the caller to enter the message, or the handshaking protocol to enable reception of messages from a data entry device. When a call is received, the controller 204 references information stored in the subscriber database 208 to determine how the message being received is to be processed. The subscriber data base 208 includes, but is not limited to such information as addresses assigned to the data communication receiver, message type associated with the address, and information related to the status of the data communication receiver, such as active or inactive for failure to pay the bill. A data entry terminal 240 is provided which couples to the controller 204, and which is used for such purposes as entry, updating and deleting of information stored in the subscriber data base 208, for monitoring system performance, and for obtaining such information as billing information.

Detailed Description Text (52):
As discussed above, a portion 1610 of the data of a message can be transmitted as a message packet. Subsequent message packets can be transmitted in the same frame or in subsequent frames of the transmission protocol, or even in subsequent frames of an alternative phase (in a multi-phase communication protocol) as will be further discussed below. The continue flag 1702 can remain set to one, for example, while there are more message packets to be received by the communication receiver 106. The final message packet is then indicated when the continue flag 1702 is set to zero. Moreover, each message packet in the sequence is identified by a message packet number 1704, which preferably increments in a circular fashion to indicate the sequential relationship between the message packets being received from frame to frame. For example, FIG. 18 is a timing diagram illustrating one such sequence of packet numbers for a transmitted message. The first message packet in the sequence may be flagged with the message packet number field 1704 being set to "11" 1802, and with the continue flag 1702 being set to 1. While there are more message packets to be received by the communication receiver 106, the continue flag 1702 will continue to be set to 1 in the first data code word 1612 of each of the subsequent message packets 1610. Further, the message packet number field 1704 can increment using modulo three arithmetic 1804 for each subsequent message packet. That is, the message packet number field 1704 for the second, third and fourth successive message packets will have the values "00", "01", "10 " 1804, with the message packet numbering sequence repeating itself in subsequent message packets. The "11" state of the message packet number field 1704 is skipped in this numbering sequence to avoid confusion with an initial packet of a non-continued message. Of course, the final message packet is indicated by the continue flag 1702 being set

to zero. Additionally, the signature field 1706 identifies all the sequential message packets as being part of one message. Hence, multiple fragmented messages can be contemporaneously received by the communication receiver 106 with the signature field 1706 of each message packet identifying the message that the particular message packet belongs to. Therefore, the combination of the continue flag 1702, the message packet number 1704, and the signature field 1706 allows the communication receiver 106 to receive and decode message fragments constituting two or more different messages being received over the same time interval. The communication receiver 106 can then join the message packets corresponding to each of the two or more messages to reconstruct each of the received and decoded messages to their original data content.

Detailed Description Text (55):
After the first word decoder 1904 receives the corrected data code word 1612 from the error corrector 1906 it checks the continue flag 1702 and the message packet number field 1704 to determine if this is a message packet of a fragmented message or if it is the one and only message packet of a complete message. If the continue flag 1702 is set to one it indicates that this message packet is part of a fragmented message. Further, if the message packet number 1704 is set to "11" then this is the first message packet in a sequence of message packets constituting a fragmented message. The message is uniquely identified by the signature field 1706. Additionally, the phase assignment field 1708 can include information to identify a subsequent phase for receiving one or more subsequent transmitted message fragments. Also, the frame assignment field 1709 can identify one or more alternative frames for receiving any subsequent transmitted message fragments. Note however that if this message packet is the one and only message packet in a message, then the continue flag 1702 will be set to zero and the message packet number 1704 will be set to "11", and the communication receiver 106 can ignore the information in the phase assignment field 1708 and frame assignment field 1709.

Detailed Description Text (58):
Alternatively, after a successful error correction the message data word decoder 1910 strips the 21 least significant bits (LSB) of the corrected data code word 1612 and couples these bits of information to the controller 816. The controller 816 then couples the information bits to the message block 1912 in the message memory 850 to append the 21 bits of information to the information already present in the first location 1914 in the message block 1912. The message data word decoder 1910 continues to decode the subsequent data code words of the data portion 1610 of the message in the current frame while enabled 1908 by the controller 816. Consequently, the 21 bits data field 1710 of each of the data code words in the data portion 1610 of the message in the current frame are coupled to the message memory 850 and appended to the information bits in the first location 1914 of the message block 1912.

Detailed Description Text (59):
After all the data code words are decoded for the data portion 1610 of the message in the current frame, the first location 1914 in the message block 1912 contains the respective message data. This message data stored in the message block 1912 can be organized to allow the controller 816 to subsequently couple the information to the display driver 840 for displaying on the display 842 in a manner well known to those skilled in the art. If the data portion 1610 of the message in the current frame constitutes the entire message then the information in the location 1914 in the message block 1912 can be retrieved by the controller 816 for presentation to a user, such as via the display 842. However, if the information in the location 1914 constitutes only a message packet of a fragmented message, as indicated by the continue flag 1702 and the message packet number 1704 in the first 21 bits of information in the location 1914, then the controller 816 will continue searching for message packets of the fragmented message until all the subsequent message packets can be received and decoded and stored in subsequent locations 1916 in the message block 1912. FIG. 19 shows for example a message block 1912 comprising the

message information from five packets of a fragmented message.

Detailed Description Text (60):
As the first data code word 1612 of each successively received message packet 1910 is decoded by the first word decoder 1904, the respective continue flag 1702, the message packet number 1704, the signature 1706, the phase assignment 1708, and the frame assignment 1709, and other included information as necessary, are coupled to the controller 816. The controller 816 then compares the signature 1706 to the stored signature in the first location 1914 of each message block 1912 in the message memory 850. A matching signature 1706 indicates that the current message packet corresponds to the fragmented message being stored in sequential packets in the message block 1912. The controller 816 then determines whether the current message packet being decoded is in the proper sequence to the already stored message packets in the message block 1912. As discussed earlier, the sequence is indicated by the message packet number 1704. If the message packet is out of sequence, this may indicate that a previous message packet was lost during transmission. The controller 816 can continue to store the successive message packets in the message block 1912 until the final message packet is decoded and stored in the message block 1912. The final message packet is indicated by the continue flag 1702 being set to zero in the first data code word 1612 of the particular message packet 1610. In this way, the communication receiver 106 can receive and decode one or more fragmented messages.

Detailed Description Text (61):
Once the communication receiver 106 detects the first message packet of a fragmented message, as indicated by the continue flag 1702 being set to 1 and the message packet number 1704 being set to "11", the communication receiver 106 will continue searching through the current frame and into subsequent frames for successive message packets corresponding to the fragmented message being received. Specifically, the communication receiver 106 searches the address field 1604 of each successive frame, or each assigned frame, for the particular address code word 1605, and then utilizes the vector code word 1607 corresponding to the address code word 1605 to point via 1609 to the next message packet 1610. The message packet is indicated by the information bits in the 21 bit data field 1710 of the first data code word 1612 of the data portion 1610 of the message in the current frame.

Detailed Description Text (65):
To indicate the amount of delay time each message or message fragment has experienced while being delayed in the delay message buffer 2010, each message or message fragment is associated with a message carry-on value stored in the delay message buffer 2010. This carry-on value can be set to a predetermined number of frame transmission cycles, e.g., thirty one cycles, which the message can be delayed with no consequence in the communication protocol. As each opportunity to transmit the message or message fragment arrives and possibly passes without transmitting at least a fragment of the message the carry-on value is decremented for the particular message to keep track of the delay time. When the carry-on value reaches zero, such as may be possible during a very busy time interval for a communication system 100, the priority of the message is elevated requiring at least a minimal message fragment to be transmitted during the next transmission frame. This minimal message fragment comprises an address code word, a vector code word, a first data code word which includes control information such as the continue flag 1702, the message packet number 1704, the signature 1706, the phase assignment 1708, and the frame assignment 1709. This minimal message fragment also comprises a message information data code word which provides one data code word of message information to the communication receiver 106. Hence, the minimal message fragment in this example comprises four code words.

Detailed Description Text (69):
The first message packet "A1" 2110 comprises 87 codewords to fill the entire next frame. This includes three additional codewords as overhead for the address

codeword, the vector codeword, and the first data codeword which provides additional control information to the communication receiver 106. As discussed earlier for FIG. 17, the first data codeword of the message packet "A1" 2110 is configured by the frame batcher 212 to include, for example, the continue flag 1702 set to one, the message packet number 1704 set to "11", and to a unique signature in the signature field 1706. Optionally, the signature 1706 can be set to a number such as the number of messages transmitted to the communication receiver 106 during the current billing cycle, which is available from the subscriber database 208 and may be included with the message in the active page file 210.

Detailed Description Text (71):
Additionally, the frame batcher 212 appends the second message packet "A2" 2112 to the next frame buffer 2008 for transmission in the next frame. The frame batcher 212 organizes the first data codeword of the second message packet "A2" 2112 to include the continue flag 1702 set to zero, the packet number set to "00" and the signature 1706 set to the same value as was included with the first message packet "A1" 2110. The frame batcher 212 then couples the message information in the next frame buffer 2008 to the next frame message buffer 2016 for subsequent processing and transmission. In this way, the terminal 102 has transmitted the original long message "A" 2108 by transmitting two smaller message packets 2110, 2112, to fill each transmitted frame as much as possible with message information, while allowing other message communication such as the new message "B" 2112 to be transmitted in its designated frame number. That is, the longer message 2108 may be slightly delayed for a full message transmission, while allowing the smaller ongoing messages 2112 to be transmitted in their designated frames.

Detailed Description Text (72):
Notice that in this example the communication receiver 106 will receive in its designated frame the first message packet "A1" 2110 which in the first data codeword identifies itself as a first message packet to the communication receiver 106. The communication receiver 106 then sequentially searches the address field of subsequent frames until it detects the second message packet "A2" 2112 whose first data codeword identifies to the communication receiver 106 that this is the final message packet. That is, the message has been completely received. If there were more message packets to be received, then the continue flag 1702 would remain set to one which would indicate to the communication receiver 106 to continue searching subsequent frames for the successive message packets.

Detailed Description Text (96):
If there are no long messages then the frame batcher 212 sets, at step 2718, the next frame carry on value to 3. The frame batcher 212 then selects the longest message in the next frame buffer having a message carry-on value not equal to zero and then transfers the message to the delay message buffer 2010 and sets the message carry-on value to 2, at step 2720. Next, the frame batcher 212 verifies, at step 2722, whether the capacity of the next frame is still exceeded by the number of codewords in the next frame buffer 2008. As long as it is exceeded, the frame batcher continues to move the longest message from the next frame buffer 2008 to the delay message buffer 2010, at steps 2720 and 2722. A frame counter is utilized by the frame batcher 212 to keep track of the number of message fragments to be transmitted in the next frame. If the next frame counter equals zero, at step 2724, then there are no message fragments in the next frame buffer 2008 and the frame batcher 212 can then transfer, at steps 2712 and 2714, the contents of the next frame buffer 2008 to the frame message buffer 216 for transmitting the next frame. On the other hand, if the next frame counter 2012 does not equal zero, at step 2724, then the frame batcher 212 retrieves the last message fragment, at step 2726.

Detailed Description Text (105):
The communication receiver 106 checks the continue flag of a received message packet 2918, and then compares the packet number to the value "11" 2920 to

determine if this is a start of a new fragmented message. If it is a new start, a new timer is created 2922 which keeps track of the maximum allowed time between receiving message packets of a fragmented message. The timer is then set to 60 seconds 2924, for example, to prevent the communication receiver 106 from remaining turned on for greater than 60 seconds while waiting for a subsequent message packet of a fragmented message.

Detailed Description Text (106):
The communication receiver 106 saves the signature and the packet number as parameters for the message in memory 2926 and then transfers 2928 the data portion of the message to message memory 850. Optionally, The controller 816 utilizes the signature and the packet number to match 2930 the received message packet with any previously received message packets that are stored in the message memory. These message packets are stored in the message memory 850 for reconstructing the fragmented message when all the message fragments have been received. The communication receiver 106 then continues 2902 searching for other transmitted messages at the next occurrence of an assigned frame 2904.

Detailed Description Text (107):
If the continue flag is not set 2918, then this indicates either that this is the very last packet in a fragmented message 2932, 2934, 2936, 2938, or that it is only a message packet of a short message 2932, 2942. In either of these cases the communication receiver 106 acknowledges receipt of a complete message by generating an alert signal and providing an alert 2940 to the user to indicate that a message has been completely received. Again, the communication receiver 106 can continue 2902 performing other functions or searching for other transmitted messages at the next occurrence of an assigned frame 2904. Therefore, the communication receiver 106 is capable of receiving fragmented messages which comprise one or more message packets over multiple frames.

L20: Entry 15 of 17 File: USPT Jun 13, 1989

DOCUMENT-IDENTIFIER: US 4839640 A

**** See image for Certificate of Correction ****

TITLE: Access control system having centralized/distributed control

Application Filing Date (1):
19880317

Brief Summary Text (5):
Alternatively, synchronized remote units allow entry to be granted if other remote units fail. However, the flexibility of a practical distributed security system is severely limited, or places a severe requirement on the communication system to be fast and accurate with the data transferred therein. Moreover, the redundancy of data stored among individual remote units for backup is relatively low, or if provided, causes significantly increased costs. Moreover, the format of the interconnection and synchronization of the various access control system elements to transfer data presents a problem to the structure of the communication channels, since the various system elements may operate somewhat independently. Furthermore, if the remote units are communicating to a central unit, the system operations and information exchange rates may be completely independent or at least different, requiring careful synchronization of the system elements or data handshaking protocol to achieve error-free data transfers.

Detailed Description Text (365):
The input FIFO interrupt routine is triggered whenever a byte comes down from the input FIFO. The routine first checks whether or not the byte is a count byte; if not, it is discarded and the system waits for another byte. If the input is a count byte, the routine waits for count data bytes to come in from the input FIFO, puts them on the input FIFO queue, and signals FIFO interrupt task that a complete message has been received. The output FIFO interrupt routine first checks whether or not the output FIFO queue is empty. If so, it sets a flag (must$tickle) and exits. If not, it outputs bytes (count and data) from the output FIFO queue into the output FIFO until the output FIFO queue is empty. Then it sets the must$tickle flag and exits.

Detailed Description Text (405):
The operator can choose to continue the system and examine the alarm messages stemming from the restore. Another restore can be initiated from the menu of commands.

First Hit Fwd Refs

Generate Collection

L22: Entry 15 of 30 File: USPT Feb 24, 1998

DOCUMENT-IDENTIFIER: US 5721743 A
TITLE: Method of and system for communicating messages

Application Filing Date (1):
19950724

Drawing Description Text (7):
FIG. 4A illustrates the transmission/reception of two concatenated messages in a frame of a batch,

Detailed Description Text (13):
FIG. 4A illustrates a transmission in a frame commencing with address code word ADD1 concatenated with message code words M1,M2,M3. Other messages beginning with respective address code words ADD2,ADD3 are concatenated with the first message. At the secondary station, its receiver is powered up for its respective frame, and, when possible, corrects any errors prior to storing all the code words that might belong to a message for it. Storing will continue, into the following frames if necessary, until a positive indication that the message for the secondary station has terminated, for example the start of a subsequent message for another station is received.

CLAIMS:

10. A method as claimed in claim 9, wherein the messages are transmitted in a predetermined frame of at least two concatenated batches, in that the decoded message or concatenated messages present in the predetermined frame of the first of the at least two batches is stored, in that the decoded message code words present in the same predetermined frame of the second and optionally subsequent batches are stored code word by code word until an indication of the end of a message is noted.

First Hit

Generate Collection

L32: Entry 27 of 127 File: DWPI Jul 9, 2003

DERWENT-ACC-NO: 2003-660068
DERWENT-WEEK: 200428

TITLE: File transmission apparatus using network, splits file into several message segments, and addresses each message segment to respective address assigned to receiving host

First Hit

Generate Collection

L32: Entry 18 of 127 File: EPAB Oct 25, 2000

PUB-NO: GB002349302A
DOCUMENT-IDENTIFIER: GB 2349302 A
TITLE: Protocol segments large message data for transmission using short message service (SMS)

PUBN-DATE: October 25, 2000

INVENTOR-INFORMATION:

NAME	COUNTRY
LEE, SANG-SEO	KR

INT-CL (IPC): H04 Q 7/22; H04 L 29/06

ABSTRACT:

First Hit

Generate Collection

L32: Entry 14 of 127 File: EPAB Nov 14, 2002

PUB-NO: WO002091658A1
DOCUMENT-IDENTIFIER: WO 2091658 A1
TITLE: PROCESSING OF SEGMENTED MESSAGES

PUBN-DATE: November 14, 2002

INVENTOR-INFORMATION:

NAME	COUNTRY
EINAMO, KARI	FI

INT-CL (IPC): H04 L 1/18
EUR-CL (EPC): H04L001/00

ABSTRACT:

CHG DATE=20030114 STATUS=O>According to the present invention, a message is sent from a first network element in a mobile communication network to a second network element in several segments, using an application protocol. To the last segment of the message a parameter is added, which indicates the number of segments of the message. At the second network element, received message segments are counted, and the number of counted message segments is compared with the number indicated by the parameter. If a mismatch of the numbers is detected, the message is discarded at the second network element.

First Hit

Generate Collection

L35: Entry 4 of 4 File: DWPI Sep 25, 1998

DERWENT-ACC-NO: 1998-575229
DERWENT-WEEK: 199849

TITLE: Conversion control system of receiving confirmation frame - transmits receiving confirmation frame message to CPU from receiver using controller after completing verification of multiple data frames and confirmation frame

Basic Abstract Text (2):
A calculation unit (21) checks the received data frames and the sequence of frames using the confirmation frame stored in the register. When the verification of a set of n data frames and the confirmation frame is completed, a receiving confirmation frame message is transmitted to the CPU from the receiver using a controller (50).

PF Application Date (1):
19970314

First Hit Fwd Refs

Generate Collection

L8: Entry 6 of 6 File: USPT Feb 2, 1999

DOCUMENT-IDENTIFIER: US 5867688 A
TITLE: Data acquisition and retrieval system with wireless handheld user interface

Application Filing Date (1):
19940214

Drawing Description Text (14):
FIG. 12 illustrates the processing sequence by which the communication server parses through a short header structure within an incoming packet to build the input buffer;

Drawing Description Text (15):
FIG. 13 illustrates the processing sequence by which the communication server parses through a long header structure within an incoming packet to build the input buffer;

Detailed Description Text (22):
The patient inquiry screen 74 also includes a virtual keypad 76 which allows the user to enter the name of a desired patient. As the user selects each letter of the desired patient's name, the CPU 34 automatically performs a search upon the entered letters and updates the scroll text window 78 correspondingly. As the user enters additional letters, the CPU 32 concatenates these letters to the end of the search string it uses and again updates the scrolling text window 78. The patient inquiry screen 74 also includes function buttons 64 along a bottom thereof to allow the user to automatically jump to an alternative screen. A scan button 77 is included to allow the user to read bar code data from a patient's wrist band, such as patient ID.

Detailed Description Text (27):
By way of example only, when the CPU 32 requests patient vitals, it expects the returned data to include, in a preset order, the patient's social security number, date on which the patient vitals were last updated, and the most recent systolic, diastolic, pulse, temperature and respiratory values. The CPU 32 sets up fields within the working space 202 for each of these values. When the returned data is received, the CPU 32 parses through the returned packet and assigns bytes therefrom to the desired field in the working space 202.

Detailed Description Text (37):
If the keypad 76 is touched, processing flows to block 1730, at which the letter is identified which corresponds to the region touched (step 1742). This selected letter is added to a temporary patient searching string within the work space memory 102 (step 1744). This letter is added to a search string (which is empty until the first letter is selected). Thereafter, the processor conducts a search based upon the search string into the patient list to identify the name most closely corresponding to the letter(s) selected from the virtual keypad 76 (step 1746). If a search string exists (i.e., the user has already entered some letters) the newly selected letter is concatenated onto the search string and a new search is conducted upon the text strings within the patient list to find the first text string which is greater in alphabetic value than (i.e., closest to) the search string. The pointer is set to the closest patient name and the scrolling text

window 78 and the scrolling bar 80 are updated (steps 1746 and 1748). If the user wishes to delete a letter from the search string, he/she simply pressing the delete region key.

Detailed Description Text (61):
The communications server 12 further includes a message buffer 110 which is utilized to store the actual data for each complete message sent from the hand held interface once all of the necessary packets have been transmitted by the handheld interface 8 and reassimilated by the CPU 106 into a single message. The COM.sub.-- INFO stored in the input buffer 108 includes a pointer MSG.sub.-- From.sub.-- HH into the message buffer to the beginning of the corresponding data string. Once every packet 300 for a message 302 is received and the corresponding data is stored in the message buffer and the COM.sub.-- INFO is stored in the input buffer 108, the CPU 106 constructs a message list therefrom. Each message list to be processed by the command server 14 is stored on one of a process queue 114. The message list includes a pointer to the corresponding data string in the message buffer 110. Each message list received from the command server 14 is initially stored in a transmit queue 116 prior to being converted back into packets 300 and transmitted to the handheld interface 8. The message list includes a pointer to a corresponding data string which is stored in the message buffer 110 when it is received from the command server 14.

Detailed Description Text (65):
The Packet Number is a long integer which is incremented each time a packet is appended to a message on the input buffer 108. The User.sub.-- ID.sub.-- To is a 6 byte value added by the handheld interface 8 to identify a destination user. If set to zero, the destination is the communications server. The communications and command servers determine which server to send the command to. This value if non-zero will identify the user of a handheld interface 8 desires to communicate therewith. The User.sub.-- ID.sub.-- From is a 6 byte value added once at login time. This user ID is assigned to the channel to identify who is logged in at that channel. This value is maintained until the person signs out. Thus, the user ID will only be transmitted once during a login session. The communications server 12 keeps track of each User ID signed onto handheld interfaces served by that communications server 12. The message total length (MSG.sub.-- Total) is a value assigned by the handheld interface 8 or by the command server 14 when a message is transmitted. The message length (MSG.sub.-- LEN) is a value updated by the short and long header parsing functions to keep track of the length of a message in the message buffer 110 as the packets for the message is added thereto. The message length field is initially tested by the communications server 12 when processing each packet in the temporary buffer 104 to determine if the packet is in a long or short header form. The message pointer field (MSG.sub.-- From.sub.-- HH) is a pointer into the message buffer to the location of the actual data message. The MSG.sub.-- From.sub.-- HH pointer is updated each time a new packet is appended to a message.

Detailed Description Text (69):
If the packet contains the long header structure, processing moves to step 718 in which the parse long header function is invoked. If the packet includes the short header structure, processing moves to step 720 where the parse short header function is invoked.

Detailed Description Text (70):
FIG. 12 and FIG. 13 illustrate the parse long and short header functions. Within the parse long header function, the first four bytes of the data packet within the temporary queue is read as the command (step 722). This command is compared with a list of valid commands and if the command is invalid the processor takes the necessary corrective action (step 724). If the command is valid, the command is written to the command field within the element of the input buffer 108 indexed by the corresponding channel number (step 726). Next, the subsequent six bytes of the

packet within the temp buffer is read as the user ID, if present, of the destination user for the attached message/data. This six byte ID is stored in the user ID.sub.-- To field of the element within the input buffer 108 indexed by the current channel number (step 728). The User ID, corresponding to the signed on user, is added to the UserID.sub.-- From field of the indexed element.

Detailed Description Text (72):
A message pointer pointing to the data string within the message buffer 110 is stored within the message pointer field of the indexed element within the input buffer 108 (step 738). Finally, the element of the temporary buffer 110 is cleared (step 740). In this manner, steps 718-740 parse through a packet having a long header structure, create the COM.sub.-- INFO structure within the input buffer 8 and store the message 302 from the handheld interface 8 within the message buffer 110.

Detailed Description Text (131):
in the case that the user picks A-Z, that letter is concatenated onto the search string which is initially empty, then a sequential search upon the text strings is performed to find the first text which is greater in alphabetic value than the search string, then that value is chosen via the function set.sub.-- scroller.sub.-- value.

Detailed Description Text (159):
place.sub.-- message--this routine will determine if the message being read in is the first or subsequent message. If the message is the first packet, then the routine parse.sub.-- long.sub.-- hdr is called. If the packet is the second or subsequent message, the routine parse--short.sub.-- hdr is called.

Detailed Description Text (160):
parse.sub.-- short.sub.-- hdr--this routine is used to parse the structure tmp.sub.-- buffer, validate the command being sent along with the correct packet number, then place the data in the appropriate in buffer message structure. This routine is called after the initial command packet is sent.

Detailed Description Text (161):
parse.sub.-- long.sub.-- hdr--this routine is used to parse the structure tmp.sub.-- buffer, validate the command being sent. This routine then initializes the in.sub.-- buffer structure and finally places the data in the in.sub.-- buffer messages structure. This routine is called when a handheld device sends its initial packet for requesting a command.

CLAIMS:

2. The system of claim 1 wherein the input computer comprises:

a command server for managing the local database; and

a communications server for receiving and transmitting packets of information to and from the portable computer, the packets being constructed in a first format having a header and a data segment, the communications server converting the packets to a second format and constructing a message therefrom, the communications server transmitting the message to the command server which returns a message list, the communications server converting the returned message list to the first format and transmitting a packet of information to the portable computer.

29. The system of claim 28 wherein the input computer comprises:

a command server for managing the local database; and

a communications server for receiving and transmitting packets of information to

and from the portable computer, the packets being constructed in a first format having a header and a data segment, the communications server converting the packets to a second format and constructing a message therefrom, the communications server transmitting the message to the command server which returns a message list, the communications server converting the returned message list to the first format and transmitting a packet of information to the portable computer.